1. Java is an object oriented programming (OOP) language.

**OOP**: a style of programming that incorporates the four features of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,   
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**Polymorphism** - a \_\_\_\_\_\_\_\_\_\_\_\_\_ declaration can refer to a \_\_\_\_\_\_\_\_\_\_\_\_\_ object. When the program actually runs, it will run according to the directions in the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Ex)

Casting: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ex)

**Abstract class**:

* Is used to define a class that will be used only to build new classes.
* Is used when you know quite a bit about an \_\_\_\_\_\_\_\_\_\_\_\_ and what you want the object to do, but yet there are still a few unknowns.
* Has at least one abstract method.
* Any subclass that \_\_\_\_\_\_\_\_\_\_\_ a super \_\_\_\_\_\_\_\_\_\_ class **must** **implement** \_\_\_\_\_\_ methods defined as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ in the \_\_\_\_\_\_\_\_\_\_ class.
* If a class has **at least** one abstract method, you **must declare the class as an abstract class** by writing the \_\_\_\_\_\_\_\_\_\_\_\_ keyword on the class header.
* When you define an abstract method, you **must** put a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ at the end of the method header. You can also write \_\_\_\_\_\_\_\_\_\_\_ keyword before static or the return type in the method header.

Ex)

**Interface**: a list of what methods are available. It has \_\_\_\_\_\_\_\_ abstract methods.

Ex)

\_\_\_\_ 2. Which of the following statements about constructors is NOT correct?

A) A constructor must have the same name as the class name.

B) A call to a constructor must always have arguments.

C) A constructor initializes the private data of an object.

D) A class can have more than one constructor.

\_\_\_\_ 3. Consider the following incomplete classes:

|  |  |
| --- | --- |
| public class Pet  {  public Pet (int a)  {  /\* implementation not shown \*/  }  // instance methods not shown  } | public class Dog extends Pet  {  public Dog()  {  *<code omitted>*  }  // instance methods not shown |

Which statement, replacing *<code omitted>*, compiles without errors?

a. super( );

b. super(4);

c. super(a);

d. super(a, 1, Display.NORTH; Display.INFINITY);

e. It is impossible to know unless you look at the API.

\_\_\_\_ 4. The \_\_\_\_ operator is used to instantiate an object from a class.

a) add

b) create

c) new

d) construct

\_\_\_\_ 5. Which of the following statements about objects is correct?

a) When programmers send commands to an object of a class, they must understand only how the object stores its private data.

b) When programmers send commands to an object of a class, they must understand only how the object's methods are implemented.

c) When programmers send commands to an object of a class, they must understand both how the object stores its private data and how its methods are implemented.

d) When programmers send commands to an object of a class, they do not need to know how the object stores its private data or how its methods are implemented.

\_\_\_\_ 6. If a class is named Student, then the constructor is named \_\_\_\_\_\_\_\_\_\_\_

a. any legal Java identifier

b. any legal Java keyword

c. StudentConstructor

d. Student

e. the default constructor

\_\_\_\_ 7. Which of the following statements about objects is correct?

a) An object defines only the methods for a class.

b) An object is a sequence of instructions that performs a task.

c) All entities, even numbers, are objects.

d) Every object has its own set of data and a set of methods to manipulate the data.

\_\_\_\_ 8. Which of the following statements about classes is correct?

a) A class is an object that can be manipulated by a program.

b) A class defines a set of objects with the same behavior.

c) Class is another name for a method.

d) A class can contain only methods.

9. Complete the class Car:

public class Car

{

private String myModel;

private int numOfSeats;

public Car(String model, int n)

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

public \_\_\_\_\_\_\_\_\_\_\_\_\_ getModel()

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ myModel;

}

public \_\_\_\_\_\_\_\_\_\_\_\_\_ drive(int m)

{

System.out.println(“It moved “ + m + “ miles.”);

}

}

10. Refer the class Person below. Then, write a subclass Student.

public class Person

{

private int myAge;

public Person()

{

myAge = 15;

}

public int getAge()

{

return age;

}

}

Student class has

* one additional private field: int myGrade
* one argument constructor: public Student(int grade)
* one accessor method: getGrade()
* and, one instance method: showData() which prints age and grade.  
  (Hint: System.out.println(\_\_\_\_\_\_\_\_\_);)

11. Fill out blanks and more…

public abstract class Quiz  
{  
 private int difficulty;  
 private int numOfQuestions;  
 public Quiz()  
 {   
 difficulty = 0;  
  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 }  
 public Quiz(int d, int n)  
 {

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}  
  
 public \_\_\_\_\_\_\_\_\_\_\_\_\_ void makeup();  
}  
  
public class AbsQuiz \_\_\_\_\_\_\_\_\_\_\_\_ Quiz  
{  
 public \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_() //default constructor  
 {  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 }  
  
 //overloaded 2-arg constructor  
  
 public \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)  
 {  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 }  
  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 { /\* implementation not show \*/}  
}

public interface RetakeQuiz

{

public abstract void retake();

}

public class BBQuiz \_\_\_\_\_\_\_\_\_\_\_ Quiz \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ RetakeQuiz

{

//Write your implementation

//Use imagination and creativity